MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 3, 2016/2017

ECP1026 – ALGORITHM AND DATA STRUCTURE

(All sections / Groups)

2 JUNE 2017
9:00 a.m – 11:00 a.m
(2 Hours)

INSTRUCTION TO STUDENTS

1. This Question paper consists of 5 pages including cover page with 4 Questions only.

2. Attempt **ALL** questions. All questions carry equal marks and the distribution of marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

Question 1

(a) Consider the finite state automata (FSA) given in Figure Q1.1.

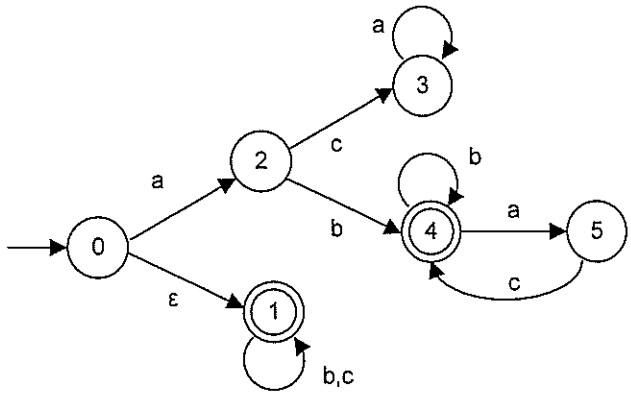


Figure Q1.1

i) Identify the set of states Q, set of input symbols Σ, initial state q_0 and set of final states F of the above FSA. [2+2+1+1 marks]

ii) Express the above FSA with a regular expression (RE). [6 marks]

(b) Draw a non-deterministic FSA diagram that accepts binary strings ending with the substring of "100". [6 marks]

(c) Consider the Turing machine (TM) given in Figure Q1.2.

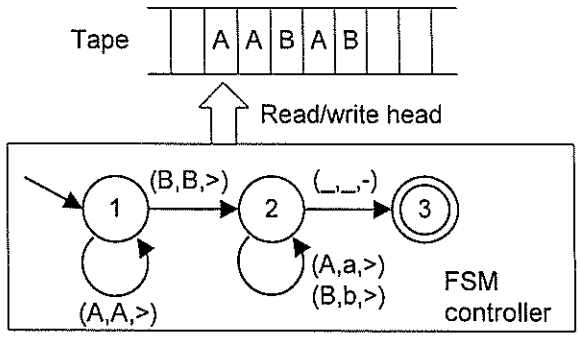


Figure Q1.2

Note: _ indicates a blank character.

Run the TM on the given input tape. During each step of the process, show clearly the current state, the contents of the tape and the character that is being read by the read/write head. [7 marks]

Continued...

Question 2

(a) Write a complete C program that reads from user an arbitrary number of records, where each record contains a student identification number and his/her CGPA. The program then displays all input records.

The program performs all the following tasks:

- Define a record consisting of an integer member `id` and a `double` member `cgpa`.
- Read in an integer value indicating number of records to be inputted.
- Create an array of record with enough memory space in a dynamic approarch.
- Input records, and assign the values into the array of record.
- Print out the contents of the array.

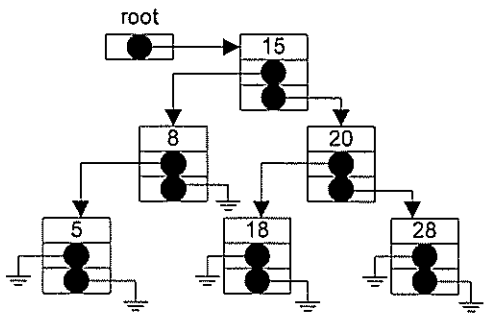
A sample output is given in Figure Q2.1. (Texts in *italic* typeface are user inputs)

```
Enter number of records: 2
Enter student id and cgpa for each record:
#1: 1011234 3.52
#2: 1011236 2.65

2 records received:
1011234 3.52
1011236 2.65
```

Figure Q2.1 [15 marks]

(b) Consider the binary tree and the record definition in Figure Q2.2.



(A) A binary tree implemented in linked-list form

```
struct Bin3Node {
  int data;
  struct Bin3Node *left;
  struct Bin3Node *right;
};
```

(B) Record definition for a node of a binary tree

Figure Q2.2

i) Assuming the function `initBin3Node` that creates and initializes a binary tree node is available. Its function prototype is given as follows:

```
struct Bin3Node *initBin3Node(int data);
```

Using `intBin3Node` and `root`, write a program segment that builds the binary tree given in Figure Q2.2. [6 marks]

ii) List the vertices of the binary tree illustrated in Figure Q2.2 as it is visited in preorder and inorder traversals. [2+2 marks]

Continued...

Question 3

(a) Consider the function $f(x,y)$ defined as follows,

$$f(x,y) = \begin{cases} y - x & \text{if } x < 0, \\ f(x-1,y) + y^2 & \text{otherwise} \end{cases}$$

i) Evaluate the result of $f(x,y)$ for $x = 2$ and $y = 1$. [2 marks]

ii) Write a recursive function based on the formula given above. The function prototype for the function is given as: `int func(int x, int y);` [7 marks]

iii) Using function `func` defined in part (a)ii), write a C statement that evaluates, and displays the result of $f(2,1)$. [2 marks]

(b) Specify the main difference between backtracking and look-ahead in terms of their area of applicability? [3 marks]

(c) Consider the C code given in Figure Q3.1.

```
1   #include <stdio.h>
2   int main(){
3     int n, k, data, sum=0, c=0;
4
5     printf("Enter number of data: ");
6     scanf("%d", &n);
7
8     for(k=1; k<=n; k++){
9       scanf("%d", &data);
10
11      if(data>=50) {
12        sum += data;
13        c++;
14      }
15    }
16
17    printf("%d passes with average %.2f\n", c, (float)sum/c);
18    return 0;
19  }
```

Figure Q3.1

Indicate clearly the time complexity of each statement, if any, and determine the worst-case time complexity and Big-O of the program. [7+2+2 marks]

Continued...

Question 4

(a) Consider the application in Question 2 part (a).

Write a function called `linearSearch` that performs linear search for a student identification number in a given array of records. The function receives an array of records, an integer indicating the size of the array and a search key, as function arguments.

The function prints the student identification number and his/her CGPA in cases where the search key matches the array element; otherwise a message "Not found" is printed. The function does not return any value.

[11 marks]

(b) A sorting algorithm is an algorithm that organizes a list of items in a certain order. Specify four factors that affect the choice of a sorting algorithm.

[4 marks]

(c) Figure Q4.1 shows the C implementation of a sorting algorithm that arranges a list of items into ascending order.

```
void unknownSort(int a[], int n){
  int i, j, k, min, curr;

  for(j=0;j<n;j++){
    min=j;
    for(k=j;k<n;k++)
      if(a[k]<a[min])
        min=k;

    curr=a[min];
    a[min]=a[j];
    a[j]=curr;

    printf("\nCycle #%d: ", j+1);
    for(i=0;i<n;i++)
      printf("%3d ", a[i]);
  }
}
```

Figure Q4.1

i) State the sorting algorithm implemented in Figure Q4.1. Explain briefly how the sorting algorithm works. [1+2 marks]

ii) Consider an array a containing the following items: 8, 25, 14, 32, 46, 22, 50, and n = 7. Write the formatted output as the function given in Figure Q4.1 is executed on a computer.

[7 marks]

End of Paper